CC41C : Introducción al Hardware
Examen–Semestre Otoño'2001
Prof.: Luis Mateu.

Pregunta 1

a. La siguiente es una secuencia de instrucciones ejecutadas bajo M32, la cual no posee *delay slots*:

```
1.- add R1,R2,R3          |        5.- cmp R7,0
2.- add R2,R4,R5          |        6.- beq 1.-
3.- sub R3,R2,R6          |        1.- add R1,R2,R3
4.- load [R6+4],R7        |        2.- add R2,R4,R5
```

Haga dos diagramas que muestren en qué ciclo se realiza cada fase de la ejecución de estas instrucciones. El primer diagrama debe considerar una arquitectura superescalar de grado 2 y el segundo una arquitectura con ejecución fuera de orden con capacidad para decodificar y retirar 2 instrucciones por ciclo del reloj. Considere que la lectura en memoria toma 4 ciclos del reloj (más los ciclos usuales de fetch, decodificación, análisis, cálculo de dirección, store, etc.).

Explique en qué momento se usan técnicas como *register bypassing*, *predicción de saltos* y *register renaming*.

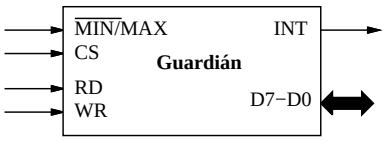
b. Se tiene un computador que posee un caché *write-back* de 4 KB con líneas de 16 bytes. Un programa realiza los siguientes accesos a memoria:

6c9a, 20f0, 78b4, 20f8, 48b4, 90f0, 20f0, 6c90, 90f8, 30f0

Las direcciones se indican en hexadecimal. En la dirección 6c9a, "a" corresponde al desplazamiento dentro de la línea etiquetada como "6c9". Además "c9" es la posición que le corresponde a esa línea en una memoria caché de 4 KB con un grado de asociatividad. Considere que los primeros 3 accesos se realizan todos en la memoria caché. Para los restantes accesos indique si se realizan en memoria caché o no, suponiendo (i) un caché con un grado de asociatividad y (ii) un caché con dos grados de asociatividad. ¡Explique!

Pregunta 2

La siguiente figura muestra un circuito *guardián* que se encarga de vigilar que la temperatura ambiente no salga de un rango dado. El guardián tiene dos puertas de E/S que permiten especificar la mínima y la máxima temperatura aceptable. Cuando la temperatura se sale de estos parámetros, el sensor activa la línea INT.



El siguiente procedimiento sirve para configurar el guardián:

```
void setTempRange(char min, char max) {
  /* min y max son enteros de un byte */
  char *port_min= (char*)0xff00;
  char *port_max= (char*)0xff01;
  *port_min= min;
  *port_max= max;
}
```

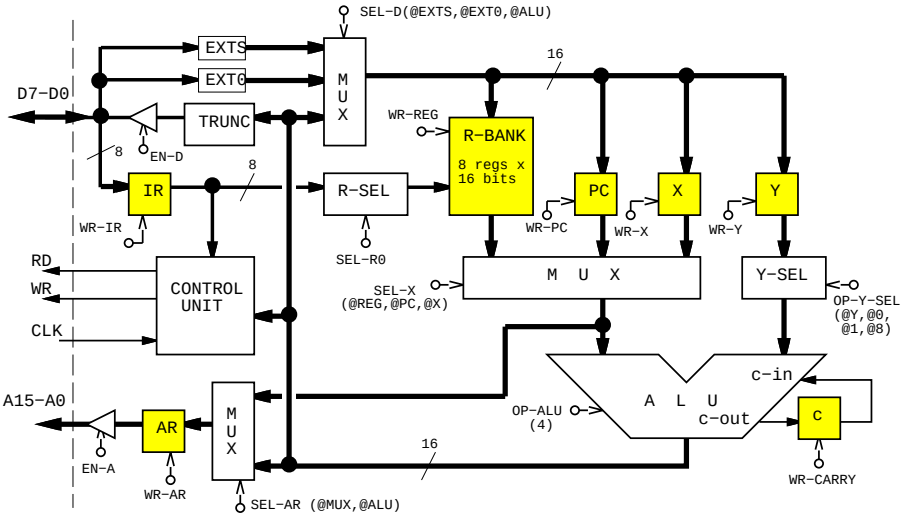
Los valores 0 y 255 se usan para indicar al guardián que no se debe producir ninguna interrupción.

a. Implemente la interfaz que se necesita para conectar el guardián con el bus del procesador, de modo que el procedimiento anterior funcione correctamente. Considere un microcontrolador con un bus de datos de 8 bits y un bus de direcciones de 16 bits.

b. Implemente en pseudo-C la rutina de atención `alertTemp` que debe procesar la interrupción causada por el guardián. Esta rutina debe desplegar un mensaje en pantalla (usando `printf`) y evitar que se siga produciendo la interrupción.

Pregunta 3

M8 es una CPU destinada a ser usada como controlador de dispositivos. En cuanto a su arquitectura lógica podemos decir que posee 8 registros de 16 bits para uso general (`R0`, ..., `R7`), tiene un contador de programa de 16 bits (`PC`), su espacio direccionable es de 64 KB y sus instrucciones son compactas, ocupando 1 o 2 bytes. Su arquitectura física se detalla en la siguiente figura:



En este diseño se observa que: los buses de datos y direcciones son de 8 y 16 bits respectivamente, el registro de instrucción IR es de sólo 8 bits y se dispone de dos registros auxiliares X e Y.

El banco de registros tiene una sola fuente de direcciones (en vez de tres): en cada ciclo del reloj se puede extraer un solo registro y al final de ese ciclo se puede actualizar ese mismo registro. Las operaciones de la ALU son las mismas de M32, pero operan sobre datos de 16 bits.

El circuito R-SEL permite seleccionar el registro que se extrae. Si SEL-R0 está en 0 se extrae el registro indicado por los últimos tres bits de IR. Si en cambio, SEL-R0 es 1, se extrae R0. El circuito Y-SEL permite seleccionar el operando Y de la ALU. Puede ser Y (`@Y`), 0 (`@0`), 1 (`@1`) u 8 (`@8`).

Se desea implementar la instrucción `addl` (*add long*) que permite sumar cantidades de 32 bits. Esta instrucción lleva como argumento explícito un registro par, por ejemplo R4, y como argumentos implícitos los registros R0, R1 y el registro consecutivo al especificado, es decir R5. Addl considera el valor de 32 bits resultante de concatenar R0 con R1 y lo suma al valor de 32 bits almacenado en R4 y R5. El resultado se almacena en R0 y R1. La instrucción se codifica en 8 bits como lo indica la siguiente figura:

a. Explique por qué el actual diseño de M8 no permite implementar `addl`.

b. Especifique qué señal de control se debe agregar a R-SEL para que se pueda implementar `addl`. Luego implemente completamente el nuevo R-SEL.

c. Utilizando el nuevo R-SEL, indique ciclo por ciclo las señales de control necesarias para ejecutar `addl` (no incluya la fase de fetch ni la fase de decodificación). Indique además para cada ciclo las transferencias entre registros que se realizan.